cleaning-sequences

October 25, 2018

1 Cleaning Sequences

There are multiple ways of sequencing the steps in your data cleaning process. We've used one so far, once in Lesson 1 and once in the **Data Cleaning Process** example video in this lesson. The **Define**, **Code**, and **Test** Markdown headers were used once in this sequence, with multiple definitions, cleaning operations, and tests under each header, respectively.

It looked like this:

1.1 Gather

```
In [1]: import pandas as pd
```

```
In [2]: df = pd.read_csv('animals.csv')
```

1.2 Assess

```
In [3]: df.head()
```

```
Out[3]:                Animal Body weight (kg) Brain weight (g)
        0  bbMountain beaver             1!35              8!1
        1               bbCow              465              423
        2         bbGrey wolf            36!33           119!5
        3              bbGoat            27!66              115
        4        bbGuinea pig             1!04              5!5
```

Quality

- bb before every animal name
- ! instead of . for decimal in body weight and brain weight

1.3 Clean

```
In [4]: df_clean = df.copy()
```

Define

- Remove 'bb' before every animal name using string slicing
- Replace ! with . in body weight and brain weight columns

Code

```
In [5]: # Remove 'bb' before every animal name using string slicing
        df_clean['Animal'] = df_clean['Animal'].str[2:]

In [6]: # Replace ! with . in body weight and brain weight columns
        df_clean['Body weight (kg)'] = df_clean['Body weight (kg)'].str.replace('!', '.')
        df_clean['Brain weight (g)'] = df_clean['Brain weight (g)'].str.replace('!', '.')
```

Test

```
In [7]: df_clean.head()
```

Out[7]:

	Animal	Body weight (kg)	Brain weight (g)
0	Mountain beaver	1.35	8.1
1	Cow	465	423
2	Grey wolf	36.33	119.5
3	Goat	27.66	115
4	Guinea pig	1.04	5.5

But you can also use multiple **Define**, **Code**, and **Test** headers, one for each data quality and tidiness issue (or group of data quality and tidiness issues). Effectively, you are defining then coding then testing immediately. This sequence is helpful when you have a lot of quality and tidiness issues to clean. Since that is the case in this lesson, this sequence will be used.

Pasting each assessment above the **Define** header as its own header can also be helpful.

Here's what this sequence looks like using the *animals.csv* dataset (and reusing the above *Gather* and *Assess* steps):

```
In [8]: # Reload df_clean with dirty animals.csv
        df_clean = df.copy()
```

1.4 Clean

bb before every animal name

Define Remove 'bb' before every animal name using string slicing.

Code

```
In [9]: df_clean['Animal'] = df_clean['Animal'].str[2:]
```

Test

```
In [10]: df_clean.Animal.head()
```

```
Out[10]: 0    Mountain beaver
         1                Cow
         2          Grey wolf
         3               Goat
         4         Guinea pig
         Name: Animal, dtype: object
```

! instead of . for decimal in body weight and brain weight

Define Replace ! with . in body weight and brain weight columns

Code

```
In [11]: df_clean['Body weight (kg)'] = df_clean['Body weight (kg)'].str.replace('!', '.')
         df_clean['Brain weight (g)'] = df_clean['Brain weight (g)'].str.replace('!', '.')
```

Test

```
In [12]: df_clean.head()
```

```
Out[12]:             Animal Body weight (kg) Brain weight (g)
         0  Mountain beaver             1.35              8.1
         1              Cow              465              423
         2        Grey wolf            36.33            119.5
         3             Goat            27.66              115
         4       Guinea pig             1.04              5.5
```

```
In [ ]:
```

```
In [ ]:
```